

REMARKS

1. Applicant thanks the Examiner for allowance of Claims 7-12 and 20-26.

5

2. Claims 1 and 13 stand rejected under 35 USC 103 as being unpatentable over Sharriit, Vlach, or Kazmierski *et al*, in view of On *et al*. The Examiner points to Sharriit, Vlach, or Kazmierski *et al* as teaching the invention as claimed except for the step of "performing a replay of the last iteration of an accepted timepoint."

10 To teach this step, the Examiner relies upon On.

Applicant respectfully disagrees.

15 Applicant will focus this discussion upon On. With regard to Sharriit, Vlach, or Kazmierski *et al*, Applicant directs the Examiner to pages 1-16 of the Specification and, in particular, to pages 10-16 of the Specification. On these pages, Applicant describes the prior art, which is embodied in the teachings of Sharriit, Vlach, or Kazmierski *et al*, as well as the shortcomings thereof that are addressed by the claimed invention. Because the Examiner concedes that these
20 cited references do not teach the invention as claimed, no further discussion of them is deemed necessary. In Applicant's view, they describe no more than Applicant had already taught was known in the art.

25 With regard to On, the Examiner is respectfully asked to consider this document's relevance to the claimed invention. The claimed invention is concerned with circuit simulation. In contrast, On is concerned with debugging software in the field of parallel programming. There is nothing within any of the cited references that suggests a motivation to combine the references.] Furthermore, a person skilled in the art of circuit simulation would have no

reason to consider On, which is concerned only with debugging parallel programs. Even if the hypothetical skilled person should consult On, there is nothing in any of the art of reference that would allow this person to adapt On's technique for debugging software to the claimed "mixed-signal behavioral models in simulation" without undue experimentation, which experimentation would clearly amount to invention. While the word "replay" appears in On, it is not concerned with a relevant teaching and the citation thereof is considered improper.

10 Specifically, On teaches:

An object of the present invention is to provide an integrated dynamic-visual parallel debugging apparatus and method for offering concentrated error debugging function with respect to an access to a shared memory between threads and message-passing interface (MPI) communication between processes where error occurrence is predicted, by providing a dynamic cyclical debugging function based upon the guarantee of an execution route, displaying a result of debugging in the form of a text, showing an execution flow of a program being debugged with a visual view, and automatically performing view mapping with respect to an execution point where error occurrence is predicted in a parallel program, while the debugging of the parallel program is performed at an SPAX.

For the accurate debugging of the parallel program within a minimum time, replaying of the execution route where errors occurred should be guaranteed and the result of the debugging with respect to the point where the error occurrence is predicted, such as the communication between processes (threads) or the access to a shared memory, should be analyzed and displayed in the various forms of visual views. Technologically

speaking, the execution of the program to be debugged should be recorded in the form of a log file, and, on the basis of the log file should be simultaneously performed both replay of the overall program and graphical view display of each debugging point. To achieve those things, a debugger should automatically map textual values, such as program variables and communication functions, into each corresponding point in a graphical view which is visually displayed on a screen during the dynamic debugging.

Accordingly, the object of the present invention is to build integrated dynamic-visual debugging environment where the result of debugging displayed as textual and graphical views is exactly searched on the basis of the execution log file and mapped into the graphical view, thereby simultaneously achieving both dynamic and visual debugging. (col. 1, lines 29-65; emphasis added)

Thus, On is concerned with an "error debugging function with respect to an access to a shared memory between threads and message-passing interface (MPI) communication between processes where error occurrence is predicted, by providing a dynamic cyclical debugging function based upon the guarantee of an execution route, displaying a result of debugging in the form of a text, showing an execution flow of a program being debugged with a visual view, and automatically performing view mapping with respect to an execution point where error occurrence is predicted in a parallel program."

Therefore, On does not disclose the claimed "method for debugging of analog and mixed-signal behavioral models in simulation."

Nor does On replay "a regular iterative equation solution process." Rather, On is concerned with "the execution of the program to be debugged [which is] recorded

in the form of a log file, and, on the basis of the log file [there is] simultaneously performed both replay of the overall program and graphical view display of each debugging point. To achieve those things, [On's] debugger ... automatically map[s] textual values, such as program variables and communication functions,
5 into each corresponding point in a graphical view which is visually displayed on a screen during the dynamic debugging." Thus, On records a program during execution and logs errors, which errors can be replayed. In effect, On runs the program with the debugger and animates the errors in the program. This is completely different from the claimed invention, which runs a circuit simulation
10 and allows a user to replay the simulation from an accepted timepoint, and during this replay interact with the simulation.

On this point, the following from On is noted:

15 The program debugging using the integrated dynamic-visual parallel debugger comprises two steps. A first step is a reference execution step 33. Primarily, a necessary routine for capture is applied to an event to be captured in a debugged object source program 34. This routine is offered by the event capture library 35. The modified program has a history file
20 for recording the trace of execution through a compiler 36 and library linker 37. The file is executed on the SPAX 38 to create an execution log file 39. A second step is a replay step 40. The routine referring to the execution log file is applied to the event which was an object of the capture in the debugged object source program. This is compiled under
25 the control of the replay driver 41 and the library linker is executed for link of a replay library 42, thereby creating a replay execution file 43. The dynamic debugging by the parallel debugger is implemented by examining the states of processes, values of variables, and register values at a symbol table 45 while executing the replay execution file at the SPAX

under the control of the parallel debugger core 44. The result of the dynamic debugging is displayed on a textual view window 47 through the source view browser 46. The dynamic cyclical debugging is performed through cyclical use of steps of setting a breakpoint or watchpoint, executing the program, and examining symbol values on the window. During the dynamic debugging, the textual/graphical view mapper 48 operates with respect to the constructs which are the points where error occurrence is predicted. For this operation, a textual/graphical view mapping dialog is called to the source view window which is one of the textual view windows. After setting a mapping object event, a number of a process running the event (a thread number in case of the threaded program), the number of times of calling out the event, and a desired graphical view, a user keys a mapping start button to carry out the mapping. If the mapping object event exists in the execution log file, the graphical view browser 49 graphically displays the execution of the program till the point where the current mapping event exists on the selected graphical view 50. So, the user can animate the execution of the program on the graphical view or continue the dynamic debugging on the textual view window. (col. 5, lines 1-41; emphasis added)

Thus, the user can watch execution of the debugged program and can animate same. The user cannot truly interact with the debugger. In particular, because the user is merely an observer of the execution of the program being replayed, the user cannot do anything to modify the flow of execution. The two step process of On is thus entirely different from that of the claimed invention. On runs a debugger to build an error log file and then allows a user to watch the program at the points at which errors are found by the debugger, *i.e.* the user can inspect and/or visualize the debugger process. The claimed invention runs a circuit simulation to obtain a solution, then it allows a user to replay the

simulation while interacting with it. In On, a user is merely replaying and observing a captured series of events, and cannot in any way alter the eventual outcome of those events.

5 Thus, yet another claim limitation is not found in On or any of the other references cited, *i.e.* that "a user only gets to interact with the simulation during said iteration replay." Nowhere in the art of record is it taught to limit user interaction to a replay of a circuit simulation. In fact, On does not concern user interaction during the replay he teaches. Rather, On animates a debugging run
10 to show the user where errors occurred during debugging of a program. The user can view the results of a debugging operation with On by replaying the debugging sequence from a log file. This is different than allowing a user to interact with a simulation while it is being replayed. On merely allows passive review of a program debug log. This is entirely different than the claimed user
15 interaction with a simulation during replay of the simulation.

In summary, On is not applicable prior art because:

- 20 • it is in an unrelated field of endeavor (program debugging for the field of parallel programming),
- does not provide any teaching that would allow a person skilled in the art to adapt it to the field of the claimed invention nor any motivation to do so (circuit simulation),
- 25 • does not relate to a mechanism for replaying circuit simulations, but relates to a mechanism for animating a debug log for a parallel program, and

- does not concern user interaction during replay, but merely enables passive viewing of a debug error log.

For the foregoing reasons, the rejections under 35 USC 103 are deemed avoided
5 and withdrawal thereof is respectfully requested. Should the Examiner find it helpful, Applicant is prepared to submit declarations pursuant to 37 CFR 1.132 in support of the foregoing, and will gladly do so if so requested by the Examiner.

3. Claims 2-6 and 14-19 stand rejected under 35 USC 103 as being
10 unpatentable over Sharrit or Kazmierski *et al*, in view of On *et al*. The Examiner points to Sharrit or Kazmierski *et al* as teaching the invention as claimed except for the step of "performing a replay of the last Newton-Raphson iteration of an accepted timepoint." To teach this step, the Examiner relies upon On.

15 Applicant respectfully disagrees.

Applicant repeats the discussion above. However, in this case there is a further problem with the prior art cited because the replay referred to by On, which was discussed above as being directed to an irrelevant and non-analogous field of
20 art, clearly lacks any teaching or suggestion a replaying a Newton-Raphson iteration.


For the foregoing reasons, the rejections under 35 USC 103 are deemed avoided and withdrawal thereof is respectfully requested. Should the Examiner find it
25 helpful, Applicant is prepared to submit declarations pursuant to 37 CFR 1.132 in support of the foregoing, and will gladly do so if so requested by the Examiner.

4. In view of the foregoing, the application is deemed to be in allowable condition. Applicant earnestly solicits the Examiner's reconsideration of the

rejections under 35 USC 103, especially with the applicability of On to the
claimed subject matter, both in terms of relevance to the pertinent field of art with
regard to the skilled person standard, and with regard to the substance of the
actual techniques taught, which contrast sharply with the claimed subject matter,
5 all as discussed above. Should the Examiner find it helpful, he should contact
Applicant's attorney, Michael A. Glenn, at (650)474-8400.

Respectfully Submitted,

10



Julia A. Thomas

Reg. No. 52,283

Attorney for Applicant

15

Customer No.: 22,862